基于容器的脉冲星数据处理软件部署与测试

冶鑫晨[1,2,4]，张海龙[1,2,3,4*]，王杰[1,2,4]，张萌[1,2]，张亚州[1,2]，王万琼[1]，李嘉[1]，杜旭[1,2]

（1. 中国科学院新疆天文台，新疆 乌鲁木齐 830011；2. 中国科学院大学，北京 100049；3. 中国科学院射电天文重点实验室，江苏 南京 210008；4. 国家天文科学数据中心，北京 100101）

摘 要：随着天文观测技术的飞速发展，天文数据处理软件日趋复杂，天文数据处理环境的部署与管理日渐繁琐。利用容器技术将脉冲星数据处理环境封装成容器镜像并建立私有镜像库，科研人员可以根据数据处理需求选择镜像，实现数据处理环境的快速部署。利用实体计算机、虚拟机和容器中对脉冲星基带数据进行相干消色散处理，对比不同平台的资源利用率与数据处理效率。实验结果表明，容器的性能与实体计算机相近；在多任务并发的数据处理测试中，容器的资源分配更加合理，相较虚拟机可以提高计算资源利用效率。在新疆天文台脉冲星数据处理服务器上实现了基于容器的数据处理架构部署，设计并开发了容器管理图形用户界面，通过优化多用户登录、身份验证、数据卷挂载管理等功能，提高利用容器技术进行天文数据处理的效率。

关键词：脉冲星；数据处理；容器；虚拟化

中图分类号：P111.44　　**文献标识码**：A　　**文章编号**： 1672-7673（2023）02

脉冲星是大质量恒星演化到末期并经过核坍缩、超新星爆发后形成的产物，具有体积小、密度大、高速旋转、强磁场等特性。科研人员利用脉冲星研究宇宙诞生和演化[1]，相关研究对于推动天体物理学、粒子物理学和卫星导航等方面的发展具有非常重要的科学意义。发现第 1 颗射电脉冲星以来，其观测和理论研究都取得了飞速发展，发现的脉冲星数目也在不断增加，截至 2022 年 10 月已确认脉冲星总数达 3341 颗（https://www.atnf.csiro.au/research/pulsar/psrcat/）。

新疆南山 25m 射电望远镜（25 m NSRT）由中国科学院新疆天文台负责运行。科研人员使用这台望远镜对近 300 颗脉冲星进行了长期计时观测，并利用 Parkes、FAST 等射电望远镜获取了大量脉冲星观测数据。脉冲星数据处理常用的软件或依赖库**错误!未找到引用源。**有 FFTW，PGPLOT，PSRCAT，SIGPROC，PSRCHVIE，TEMPO/TEMPO2，DSPSR 和 PINT 等，这些软件之间相互依赖关系复杂。随着天文观测手段的发展，观测产生的数据量不断增加，桌面计算机已无法满足数据处理需求，文[5]调研中国虚拟天文台核心功能需求时认为，在线计算服务可以更高效地进行海量天文数据处理。在公共的数据服务器上进行数据处理环境部署往往需要花费科研人员大量的时间和精力，对软件版本与依赖关系的管理也是具有挑战的工作。

早期的研究尝试使用基于 Hypervisor 的虚拟机技术（以下简称“虚拟机”）实现科学数据处理环境部署及软件版本管理。文[7]使用虚拟机部署混合云，使 A&A 的数据和应用程序不受基础架构影响以相同的方式工作。虚拟机在高输入/输出和高并发的科学数据处理方面存在劣势：（1）虚拟机的虚拟化机制造成计算资源的过度消耗；（2）虚拟机跨平台移植性弱，数据处理环境难以共享；（3）虚拟机缺乏应用级的监控能力和资源调度的灵活性。

容器技术[8]有望解决科学数据处理环境的部署与管理中存在的问题。容器技术不完整的虚拟化操作系统，通过共享主机内核实现轻量级操作系统层虚拟化，利用 Namespace 和 Cgroup 实现虚拟环境的隔离与管理。文[9]评估了虚拟机和容器技术的性能，结果表明在中央处理器性能、内存吞吐量、磁盘输入/输出和负载测试方面，容器技术的表现优于虚拟机。文[10]提出使用容器技术是构建科学数据处理平台的理想选择，在隔离的执行环境中分配不同科学目标的任务，可以显著降低数据处理过程的复杂度，解决依赖库版本冲突问题。文[11]采用基于 Docker/VM 的架构，利用 Python 等编程语言实现服务器端数据批处理。文[12]使

用容器技术实现了可见度函数校准软件 SAGECaL 的快速部署，研究表明，使用容器技术实现自动部署的方法极大提高了 SAGECaL 分布式集群的部署效率。文[13]在容器环境中测试了天文软件的运行，容器技术可以帮助科研人员重新配置数据处理环境并生成测试结果。文[14]创建了基于 Docker 的容器框架（Kliko），用于运行一个或多个相关的计算作业，并实现了基于 Web 的容器调度器和专门用于天文数据输出的可视化工具。文**错误!未找到引用源。**将研究所需的软件环境封装在容器镜像中，方便其他科研人员复制和扩展结果。

容器管理通常使用命令行界面（Command-Line Interface，CLI），科研人员需要时间来学习如何利用容器处理数据，降低了科学数据的处理效率。目前已有的工具尝试使用图形用户界面（Graphical User Interface，GUI）管理容器，例如 Portainer（https://www.portainer.io/），minikube GUI（https://minikube.sigs.k8s.io/docs/tutorials/setup_minikube_gui/），DockStation（https://dockstation.io/）等。但这些工具并未针对科学数据处理进行优化，对于利用容器技术创建数据处理环境的使用场景，这些工具仍存在优化空间。

本文基于 Docker 开源容器引擎实现了新疆天文台脉冲星数据处理环境的封装与快速部署，对比测试了容器与虚拟机中进行脉冲星数据处理的资源利用率与稳定性，设计并实现了控制容器框架的图形用户界面，针对天文数据处理需求进行了功能优化。在新疆天文台脉冲星数据处理服务器上完成了容器框架的部署，应用于实际的科学数据处理工作当中。

1 容器镜像

1.1 镜像封装

为了提高科研人员数据处理的灵活性并提高资源利用率，本文将脉冲星数据处理软件和依赖分层封装在标准的容器镜像当中，各层级镜像所包含的软件如表 1。

表 1 镜像的组成

Table 1 Components of the Images

Type	Layer	The components of the Top Layer
A	Layer A	基础 Linux 系统（Ubuntu）
B	Layer A+B	libX11, gcc, gcc-c++等基础依赖环境
C	Layer A+B+C	FFTW、 CFITSIO、 PGPLOT 及其它基础软件
D	Layer A+B+C+D	TEMPO2、PSRCHIVE 等脉冲星数据处理软件
E	Layer A+B+C+D+E	Anaconda、 TensorFlow 等进阶工具

以 Ubuntu 为基础镜像，在不同层级的镜像中有计划地封装了不同对应关系的依赖环境与软件版本。例如对 PGPOLT 绘制的图像利用 Pillow 库进行处理，当需要运行编写较早的程序时，可以生成一个 Python2 的容器环境使用 PIL 库。刚接触脉冲星数据处理的研究生，可以直接使用完整的脉冲星数据处理环境快速开展科学研究，由于容器的隔离性，无需担忧配置错误影响同一服务器上运行的其他容器。有经验的科研人员可根据需求选择指定版本的软件环境，并允许通过生成多个容器组合成数据处理管线。

1.2 图形界面的容器配置

根据科研人员提出的需求，本文尝试对带图形界面的数据处理环境进行容器封装测试。天文数据处理中需要使用的依赖关系复杂，使用 DockerHub（https://hub.docker.com/）中已有的 VNC 镜像在部署脉冲星数据处理软件时出现依赖冲突，导致实现困难。因此在部署带图形界面的天文数据处理环境时，本文基于已安装好脉冲星处理软件的镜像，使用 Xfce 桌面实现图形界面，并利用 Tightvncserver 实现远程访问。实体计算机配置方面需要在主系统中安装 x11-xserver-utils，允许 Docker 访问 X11 的显示接口。在容器创建时需要共享本地 unix 端口并修改环境变量，避免在容器内运行 GUI 应用程序遇到问题。

2 基于容器的数据处理架构部署

2.1 架构设计

在新疆天文台脉冲星数据处理服务器上部署了基于容器的数据处理架构，架构设计包括硬件环境、软件部署及图形用户界面，具体设计如图 1。

2.2 硬件环境

脉冲星数据处理环境核心为数据处理服务器，数据处理服务器型号为浪潮 NF5460M4，配置如表 2。

表 2 数据处理服务器配置

Table 2 Configuration of the data processing server

Name	
CPU	Intel Xeon E5-2630 v4 *2 (20 cores 40 threads)
RAM	256 GB RAM
GPU	NVIDIA Tesla P4 8GB
Network interface	10GbE
Hard disk	1TB SSD *2 10TB HDD *36

数据处理服务器使用万兆以太网挂载了多台 NFS 服务器，扩展存储空间；通过千兆以太网连接管理服务器与公共数据服务器。将容器管理模块与私有镜像库部署在独立的服务器上有利于系统的维护，且可将研究团组的数据处理环境快速迁移至高性能计算集群。

2.3 软件部署

2.3.1 基础软件

数据处理架构部署的基础软件信息如下：

Docker_19.03.8：使用 Docker 作为容器引擎，通过中间层与本文开发的图形用户界面连接。

cAdviosr_0.43.0：cAdviosr 用于分析容器的资源使用情况和性能指标，负责收集容器中的实时信息，并确定主机的系统资源使用情况。

InfluxDB_1.5.3：存储 Cadiviosr 生成的数据，实现监控数据的持久化。

Grafana_8.3.2：通过使用可视化工具向管理员展示监控数据和用户的容器信息。

Tomcat_5.0.28：为图形用户界面提供服务的应用服务器。

Mysql_5.0.24：用于存储用户信息。

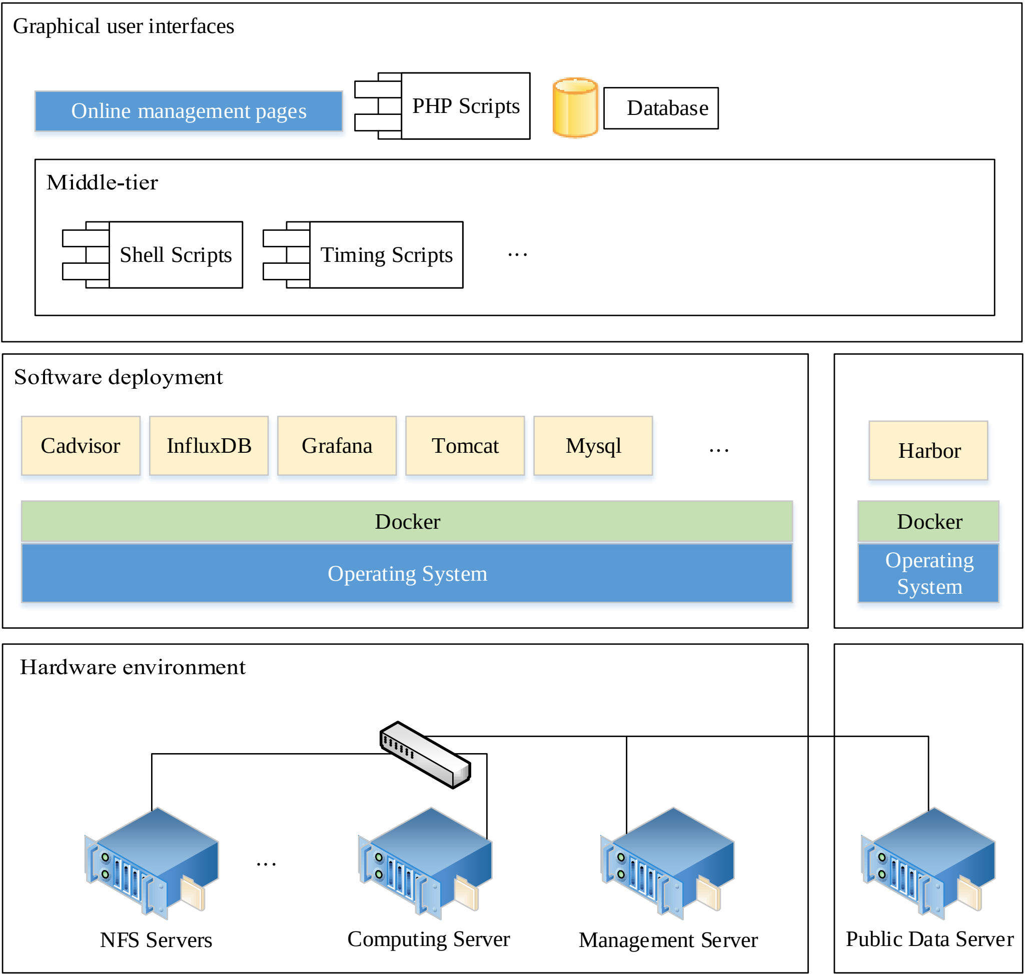


图 1 基于容器的脉冲星数据处理环境架构

Fig. 1 Container-based architecture of pulsar data processing environment

2.3.2 私有镜像库

由于网络带宽的限制与私人镜像打包的需求，本文使用 Harbor_2.3.4 私有库工具存储与管理脉冲星数据处理环境的镜像。私有镜像库部署在公共数据服务器上，由于科研人员不直接使用 Harbor 对镜像进行操作，实际的镜像使用权限由图形用户界面控制。

2.4 图形用户界面

科研人员可登录图形用户界面使用对用户信息、镜像、容器、数据卷进行管理，通过页面操作指引，科研人员无需学习容器的操作与实现原理，图形用户界面首页如图 2。管理页面只记录操作信息，中间层负责将操作信息转换为具体的指令控制容器框架，完成用户的操作指令。图形用户界面功能模块如图 3。

2.4.1 多用户管理与身份验证

容器框架在使用中通常由管理员集中管理，在科学数据处理中服务器通常由多个科研人员共享使用，灵活的多用户管理方式更适合具体需求。本文在图形用户界面中实现了多用户管理，科研人员可以直接管理个人的容器、镜像和数据卷。用户权限由图形用户界面根据数据库中信息进行控制，底层容器框架的升级及硬件平台的更换不会对科研人员的操作带来影响。

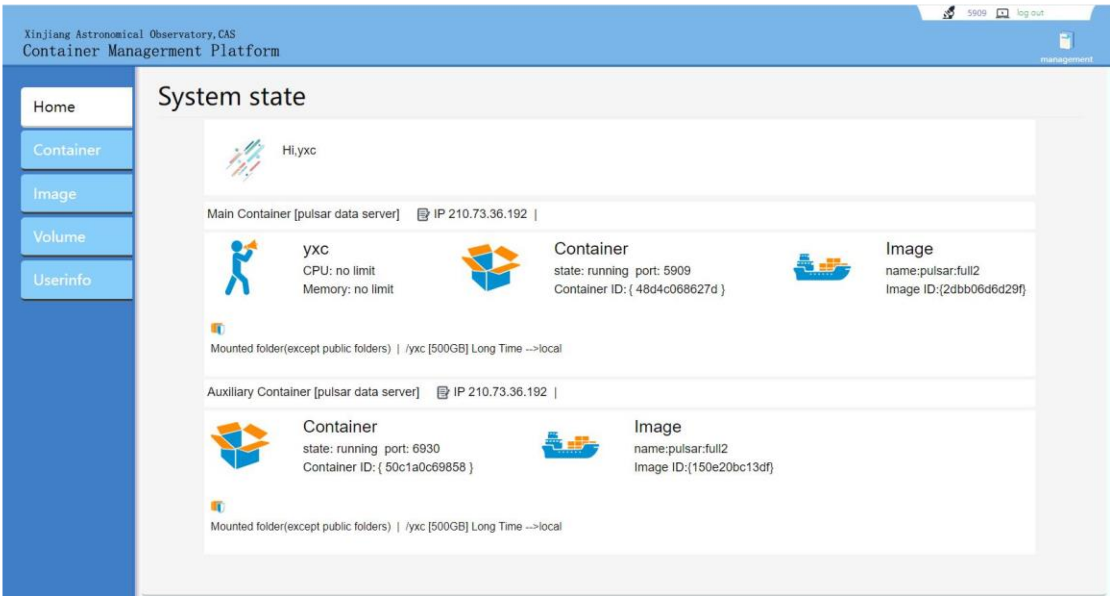


图 2 图形用户界面

Fig. 2 Graphical user interface

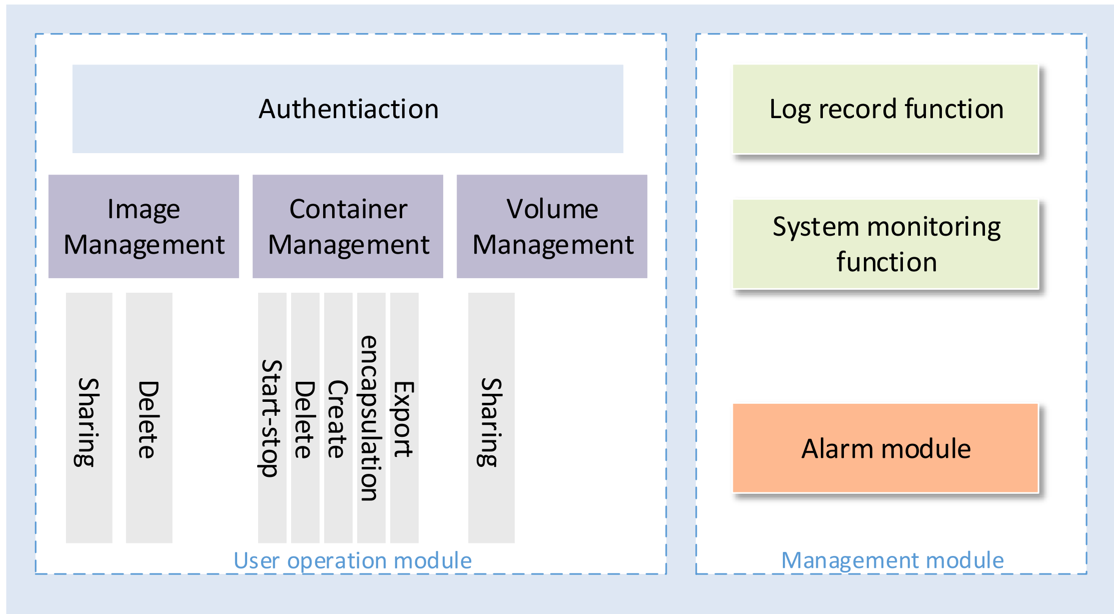


图 3　图形用户界面功能模块

Fig. 3　Functional modules of the graphical user interface

与传统的管理系统独立的身份验证方式不同，本文通过验证“初始容器”完成身份验证。系统创建新用户时同时生成一个“初始容器”，该容器包含完整的脉冲星数据处理环境，科研人员可直接使用该容器进行数据处理。当科研人员使用图形用户界面进行容器管理时，只需输入“初始容器”端口号和“初始容器”中的用户密码，身份验证将通过中间层直接在容器内进行。这种认证方式的优点是不需要记录专用的管理密码，数据库中也不存储密码，提高科研人员使用的便捷性与安全性。

2.4.2 容器管理

容器管理模块可实现启动/停止容器、删除容器、创建容器、封装容器以及导出容器。容器创建功能采用引导的方式，系统根据操作信息生成容器并记录生成信息，科研人员通过

自动分配的端口号登录容器。导出容器功能可将容器封装至.tar 文件并存储至公共数据服务器，科研人员在指定时间内利用生成的下载地址保存导出的容器。

2.4.3 镜像管理

容器管理模块可实现镜像的删除与分享。镜像分享功能可将封装后的容器分享给系统内的其他科研人员使用。

2.4.4 数据卷管理

为了在多用户环境下灵活使用容器数据卷，满足天文数据处理的需求，本文将数据卷分为 3 类：（1）只读共享数据卷; (2)读写共享数据卷;(3)个人数据卷。只读共享数据卷存储公共数据，只读属性可以有效防止误删除，只有指定的账号拥有写入权限。读写共享数据卷为两类：（1）HDD 上的读写共享数据卷，便于不同用户之间的数据传输与临时大容量中间数据的存储；（2）SSD 上的读写共享数据卷，供高 I/O 操作的数据处理使用。默认情况下个人数据卷只可本人使用，数据卷共享功能可选择只读/读写的方式分享个人数据卷给其他科研人员，便于合作与交流。

3 基于容器的脉冲星数据处理

3.1 脉冲星数据处理性能对比测试

3.1.1 脉冲星数据相干消色散

星际介质是低温的等离子体，脉冲星信号通过星际介质到达地球时会产生色散，使得高频部分先到达，低频部分后到达[15]。由星际介质导致频率v_1,v_2的观测时延Δt，$\Delta t = k_{DM} \cdot DM \cdot (v_1^{-2} - v_2^{-2})$，其中，$k_{DM}$是色散常数，大小为$4.148808 MHz^2 pc^{-1} cm^3 s$；$DM$是色散度，它是脉冲星与地球视向距离上电子数密度的积分，由$DM = \int_0^d n_e dl$ 表示，其中n_e是电子数密度，d为脉冲星与观测望远镜的视向距离。星际介质对脉冲星信号产生严重的色散影响，造成脉冲轮廓展宽和变形，严重时甚至可能导致无法观测到脉冲轮廓，因此必须对脉冲星观测数据进行消色散处理。脉冲星数据消色散的方法有两种，即非相干消色散与相干消色散。

相干消色散是将色散效应等效为滤波器，首先将脉冲星数据进行傅里叶变换，将频域上的数据乘以等效滤波器传递函数的反函数（*chirp* 函数），之后再进行傅里叶变换的逆变换，最终得到消色散后的脉冲星数据。离散 *chirp* 函数[17]为

$$chirp = T_k H_k^{-1} = \frac{1}{N}\left[1 + \left(\frac{f_k}{0.47B}\right)^{80}\right]^{-\frac{1}{2}} \exp\left\{-i\left[\frac{\pm 2\pi DM f_k^2}{(2.41\times10^{-10}) f_0^2 (f_0 \pm f_k)}\right]\right\}, \quad (1)$$

其中，$f_k = \begin{cases} \left(\frac{k}{N}\right)B & 0 \le k < \frac{N}{2} \\ \frac{k-N}{N}B & \frac{N}{2} \le k < N \end{cases}$;N 为离散信号的频域点数；$f_0$为观测的中心频率；B 为观测带宽。

相干消色散理论上可以完全消除星际介质的色散影响，但是其过程包含傅里叶变换及逆变换，需要占用大量的 CPU 资源和内存资源。

3.1.2 测试环境和测试数据

为对比在实体计算机、虚拟机和容器中脉冲星数据处理性能，本文在实验环境中对脉冲星数据进行相干消色散，测试环境信息见表 3，测试数据信息见表 4。

表 3 测试环境

Table 3 Test environment

Name	
CPU	Intel i9-10900K (10 cores 20 threads)
RAM	64 GB RAM

Software	TEMPO2 2020.04.1
OS	Debian 10.10.0
Network Interface	10GbE
Hard Disk	1TB NVMe SSD

表 4 测试数据

Table 4 Test data

Name	
Observation Target	J0437-4715
Telescope	Parkes
Polarization	2
Sampling	8 bits
Data Type	Baseband data (.dada)
Size of the Data	12.80GB

3.1.3 单任务对比测试

本文分别基于实体计算机、虚拟机和容器，利用 TEMPO2 对脉冲星 J0437-4715 的基带数据进行相干消色散处理，设置相同线程数，关闭非必要程序。**测试结果见表 5~表 7。**

表 5 实体计算机数据处理时间

Table 5 Data processing time of the physical computer

	1	2	3	4	5
Time(s)	45.043	45.148	45.103	45.107	45.098

表 6 虚拟机数据处理时间

Table 6 Data processing time of the VM

	1	2	3	4	5
Time(s)	47.783	47.623	47.975	47.674	47.902

表 7 容器数据处理时间

Table 7 Data processing time of the container

	1	**2**	**3**	**4**	**5**
Time (s)	46.028	45.962	45.847	45.991	46.084

对比测试结果可以看出，各平台测试结果相差不大，容器消耗的时间更接近实体计算机。

3.1.4 多任务对比测试

在相同配置的三台实体计算机（硬件信息见表 3）上分别运行三个、五个线程/虚拟机/容器，使用测试脚本使每轮测试中各线程/虚拟机/容器同时对数据进行相干消色散处理（模拟多用户处理场景），共测试 5 次。

在对比实验中，以最后一个线程/虚拟机/容器完成数据处理的时间为单次测试的耗时，数据处理所耗时间如图 4（a）和图 4（b）。为对比不同平台的资源分配均衡性，对测试结果计算总体标准差（σ）：

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}, \qquad (2)$$

耗时的总体标准差如图 4（c）和图 4（d）。

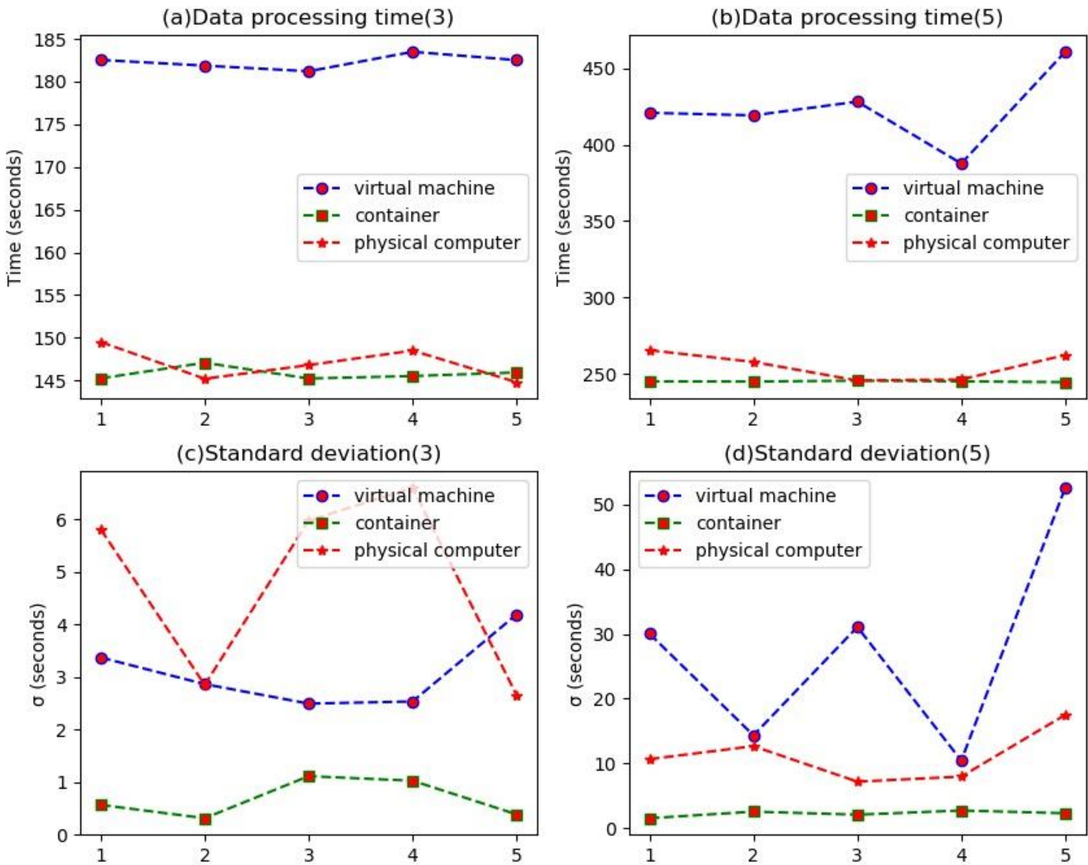


图 4 多任务性能对比测试

Fig.4 Multitasking performance comparison test

实验结果表明，容器相较于虚拟机完成相同计算量的脉冲星数据处理所耗时间更少，所耗时间波动和总体标准差对比分析结果表明容器技术的资源分配相较于虚拟机更为均衡，且数据处理过程更为稳定。与实体计算机测试结果对比表明，在多任务运行时资源的均衡分配能带来更优的总体数据处理性能。

3.2 脉冲星数据处理管线测试

使用所部署的数据处理架构（数据处理服务器配置见表 2）创建容器并实现脉冲星测时数据处理管线，测试使用 Parkes 望远镜[15]2004 年发表的观测数据，数据信息见表 8。脉冲星数据处理通常要经历以下步骤：数据下载、数据预处理、具体数据分析。根据不同步骤的数据处理需求，利用私有镜像库的镜像文件生成多个容器，实现数据的自动预处理，并利用容器内的脉冲星数据处理软件进行分析，所使用的容器信息见表 9。

表 8 测试数据

Table 8 Test data

Name	
Observation Target	J1539-5626
Telescope	Parkes
Data Type	PSRFITS(.rf)
Observation Mode	PSR
Number of Files	4

表 9 容器信息

Table 9 Containers information

Container	Image	Mounted data volumes
A	B-Type	读写共享数据卷，个人数据卷
B	D-Type	个人数据卷
C	D-Type	只读共享数据卷，读写共享数据卷，个人共享数据卷

数据处理流水线如下：

1) 由于公共服务器访问限制（指定容器暴露端口），使用容器 *A* 下载数据文件夹并转存在个人文件夹的子目录“rf”当中。
2) 容器 *B* 上运行的循环脚本负责监控个人文件夹子目录“rf”中的.rf 文件，发现新存入的.rf 文件会自动调用 PSRCHIVE 命令进行预处理：‘pam -DFTp -e FT *.rf’。
3) 容器 *B* 将预处理后的数据转存至个人文件夹子目录“ft”当中。
4) 使用容器 *C* 查看预处理后的数据文件脉冲轮廓：‘pav -DFTp *.FT’，如图 5。

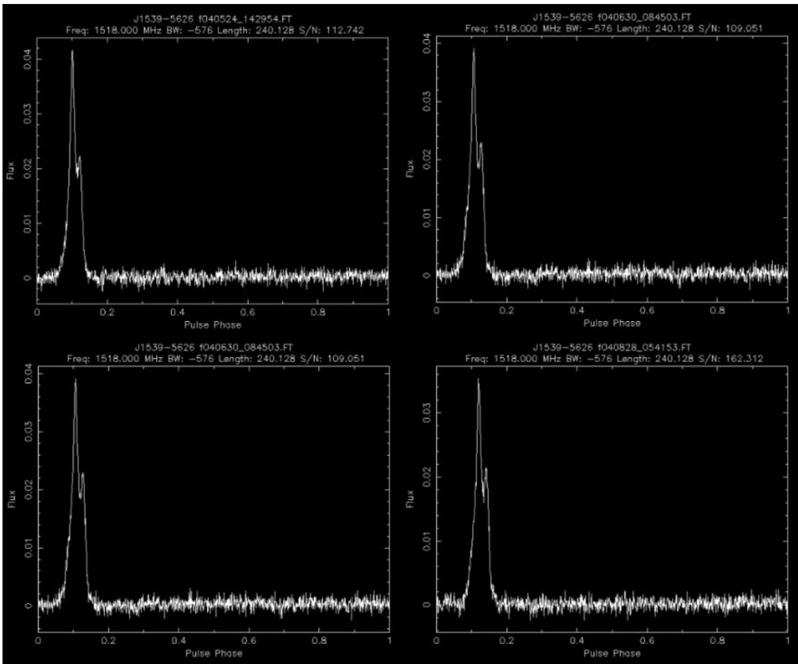


图 5 预处理后的脉冲轮廓（pav -DFTp *.FT）

Fig.5 Pulse profile after preprocessing (pav -DFTp *.FT)

5) 选取信噪比高的文件利用 PSRCHIVE 的 paas 命令制作标准轮廓（.std）。
6) 使用 PSRCHIVE 和 PSRCAT 软件生成 tim 文件和 par 文件。数据拟合后得到时间序列残差，结果如图 6。

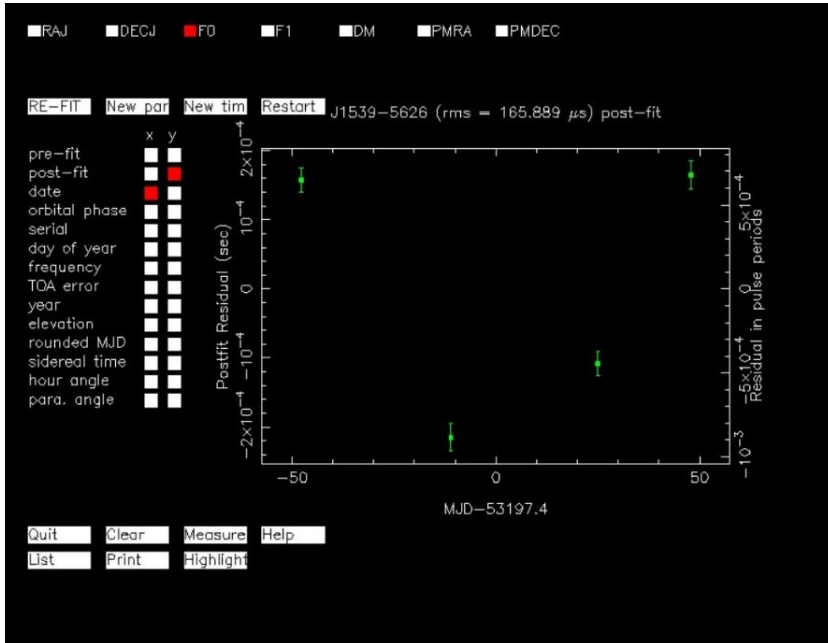


图 6 时间序列残差结果

Fig.6 Timing residual result

3.3 科研成果

FAST 观测产生的数据量巨大，桌面计算机上无法完成数据处理工作。完成容器框架在脉冲星数据处理服务器上的部署后，科研人员利用快速生成的数据处理环境来处理 FAST 数据并已产生科学成果。

黑寡妇脉冲星中发现的第 4 例等离子体透镜现象[18]。PSR J1720-0533 是在 FAST 多科学目标同时巡天（the Commensal Radio Astronomy FAST Survey，CRAFTS）[19]的漂移扫描中发现的。科研人员利用本文部署的数据处理架构对 PSR J1720-0533 观测数据进行处理，发现该脉冲星存在明显的掩食现象，这是黑寡妇脉冲星的典型特征。使用图形用户界面可以快速生成多个脉冲星数据处理环境，在各个容器内运行不同的数据处理脚本可以提高对比分析的效率，科研人员对数据处理结果进一步分析发现该脉冲星在进入掩食过程中存在等离子透镜现象，这是黑寡妇脉冲星种族中发现的第四例等离子透镜现象。

4 总 结

本文基于容器技术封装了脉冲星数据处理环境并建立私有镜像库，实现了脉冲星数据处理环境的快速部署。在脉冲星数据处理服务器上实现了基于容器的数据处理架构部署，设计并开发了图形用户界面，根据科学数据处理需要优化了容器管理功能，降低了科研人员利用容器进行数据处理的学习成本。使用实体计算机、虚拟机和容器对脉冲星数据处能进行了对比测试，测试表明容器的数据处理性能与实体计算机相当，优于虚拟机。在多任务并发的情况下，容器环境具有更好的负载均衡能力及稳定性。目前完成部署的数据处理服务器已用于脉冲星数据处理工作中并产生了相关科研成果。

参考文献:

[1] BETHAPUDI S, DESAI S. Separation of pulsar signals from noise using supervised machine learning algorithms[J]. Astronomy and Computing, 2018, 23（1）: 15-26.

[2] VAN STRATEN W, BAILES M. DSPSR: digital signal processing software for pulsar astronomy[J]. Publications of the Astronomical Society of Australia, 2011, 28(1): 1-14.

[3] LUO J, RANSOM S, DEMOREST P, et al. PINT: a modern software package for pulsar timing[J]. The Astrophysical Journal, 2021, 911(1): 45.

[4] HOBBS G B, EDWARDS R T, MANCHESTER R N. TEMPO2, a new pulsar-timing package–I. An overview[J]. Monthly Notices of the Royal Astronomical Society, 2006, 369(2): 655-672.

[5] VAN STRATEN W, MANCHESTER R N, JOHNSTON S, et al. psrchive and psrfits: definition of the stokes parameters and instrumental basis conventions[J]. Publications of the Astronomical Society of Australia, 2010, 27(1): 104-109.

[6] 许允飞,樊东卫,崔辰州,等. 中国虚拟天文台的核心功能需求调查分析[J].天文研究与技术,2020,17(1): 111-120.
XU Y F, FAN D W, CUI C Z, et al. Investigation and analysis of core functional requirements of China VO[J]. Astronomical Research & Technology, 2020, 17(1): 111-120.

[7] BERTOCCO S, DOWLER P, GAUDET S, et al. Cloud access to interoperable IVOA-compliant VOSpace storage[J]. Astronomy and Computing, 2018, 24（1）: 36-44.

[8] MERKEL D. Docker: lightweight linux containers for consistent development and deployment[J]. Linux Journal, 2014, 2014(239): 2.

[9] POTDAR A M, NARAYAN D G, KENGOND S, et al. Performance evaluation of docker container and virtual machine[J]. Procedia Computer Science, 2020, 171: 1419-1428.

[10] GERLACH W, TANG W, WILKE A, et al. Container orchestration for scientific workflows[C]//Proceedings of the 2015 IEEE International Conference on Cloud Engineering. 2015: 377-378.

[11] TAGHIZADEH-POPP M, KIM J W, LEMSON G, et al. SciServer: a science platform for astronomy and beyond[J]. Astronomy and Computing, 2020, 33: 100412.

[12] 姚坤,戴伟,杨秋萍,等. 基于容器技术的天文应用软件自动部署方法[J].天文研究与技术,2019,16(3): 321-328.
YAO K, DAI W, YANG Q P, et al. Automatic deployment method of astronomical application software based on container technology[J]. Astronomical Research & Technology, 2019, 16(3): 321-328.

[13] MORRIS D, VOUTSINAS S, HAMBLY N C, et al. Use of Docker for deployment and testing of astronomy software[J]. Astronomy and Computing, 2017, 20: 105-119.

[14] MOLENAAR G, MAKHATHINI S, GIRARD J N, et al. Kliko—the scientific compute container format[J]. Astronomy and Computing, 2018, 25: 1-9.

[15] GIOMMI P, BRANDT C H, DE ALMEIDA U B, et al. Open universe for blazars: a new generation of astronomical products based on 14 years of Swift-XRT data[J]. Astronomy & Astrophysics, 2019, 631: A116.

[16] TIBURZI C. Pulsars probe the low-frequency gravitational sky: pulsar timing arrays basics and recent results[J]. Publications of the Astronomical Society of Australia, 2018, 35: e013.

[17] STAIRS I H. Observations of binary and millisecond pulsars with a baseband recording system[D]. Princeton: Princeton University, 1998.

[18] WANG S Q, WANG J B, WANG N, et al. Unusual emission variations near the eclipse of black widow pulsar PSR J1720− 0533[J]. The Astrophysical Journal Letters, 2021, 922(1): L13.

[19] LI D, WANG P, QIAN L, et al. FAST in space: considerations for a multibeam, multipurpose survey using china's 500-m aperture spherical radio telescope (FAST)[J]. IEEE Microwave Magazine, 2018, 19(3): 112-119.

Deployment and Testing of Pulsar Data Processing

Software Based on Container

Xinchen Ye[1,2,4], Hailong Zhang[1,2,3,4*], Jie Wang[1,2,4], Meng Zhang[1,2], Yazhou Zhang[1,2], Wanqiong Wang[1], Jia Li[1], Xu Du[1,2]

(1. Xinjiang Astronomical Observatory, Chinese Academy of Sciences, Urumqi 830011, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China; 3. Key Laboratory of Radio Astronomy,

Chinese Academy of Sciences, Nanjing 210008, China; 4. National Astronomical Data Center, Beijing 100101, China)

Abstract: With the rapid development of astronomical observation technology, astronomical data processing software has become increasingly complex, and it is challenging to deploy the software environment. Container technology was used for packaging the pulsar data processing environment to images by applying hierarchical encapsulation. According to the data processing requirements, a private library of images has been established, and the users can select the images according to the data processing mode. The hardware environment, the traditional virtual machine environment, and the container environment performed coherent dedispersion for pulsar baseband data. The resource utilization and data processing efficiency of different platforms are compared. Experimental results show that the performance of the container is similar to the physical machine. In multi-threaded parallel processing, the resource allocation of the container is more reasonable and can improve the efficiency of service resource utilization. Implemented container-based data processing architecture deployment on Xinjiang Astronomical Observatory pulsar data processing server, designed and developed the container management graphical user interfaces. Through optimizing functions such as multi-user login, authentication, and data volume mount to improve the efficiency of astronomical data processing using container technology.

Key words: pulsar; data processing; container; virtualization;